ESD-TR-79-73

Semiannual Technical Summary

Information Processing Techniques Program

Volume I: Packet Speech Systems Technology

31 March 1979

Prepared for the Defense Advanced Research Projects Agency
under Electronic Systems Division Contract F19628-78-C-0002 by

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. This work was sponsored by the Defense Advanced Research Projects Agency under Air Force Contract F19628-78-C-0002 (ARPA Order 2006).

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Joseph C. Syiek

Joseph C. Syiek
Project Officer
Lincoln Laboratory Project Office

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

INFORMATION PROCESSING TECHNIQUES PROGRAM
VOLUME I: PACKET SPEECH SYSTEMS TECHNOLOGY

SEMIANNUAL TECHNICAL SUMMARY REPORT
TO THE
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

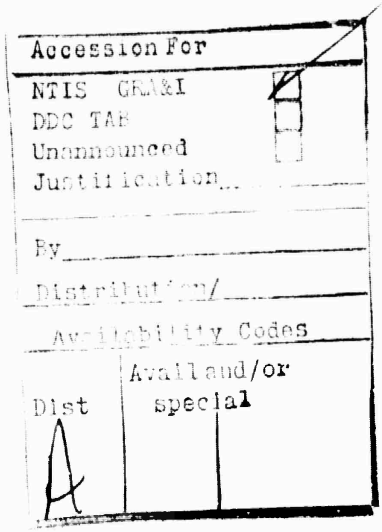
1 OCTOBER 1978 – 31 MARCH 1979

ISSUED 8 AUGUST 1979

LEXINGTON MASSACHUSETTS

ABSTRACT

This report describes work performed on the Packet Speech Systems Technology Program sponsored by the Information Processing Techniques Office of the Defense Advanced Research Projects Agency during the period 1 October 1978 through 31 March 1979.



CONTENTS

INTRODUCTION AND SUMMARY

The long-range objectives of the Packet Speech Systems Technology Program are to develop and demonstrate techniques for efficient digital speech communication on networks suitable for both voice and data, and to investigate and develop techniques for integrated voice and data communication in packetized networks, including wideband common-user satellite links. Specific areas of concern are the concentration of statistically fluctuating volumes of voice traffic; the adaptation of communication strategies to conditions of jamming, fading, and traffic volume; and the eventual interconnecting of wideband satellite networks to terrestrial systems.

Previous efforts in this area have led to new vocoder structures for improved narrowband voice performance and multiple-rate transmission, and to demonstrations of conversational speech and conferencing on the ARPANET and the Atlantic Packet Satellite Network.

The current program has two major thrusts; i.e., the development and refinement of practical low-cost, robust, narrowband and variable-rate speech algorithms and voice terminal structures, and the establishment of an experimental wideband satellite network to serve as a unique facility for the realistic investigation of voice/data networking strategies.

This report covers work in five areas: the development of a custom LSI-based narrowband channel vocoder, studies of improved vocoder structures and algorithms, the development of hardware facilities for further research and evaluation of speech systems and terminal designs, progress in satellite network and internetted voice conferencing in the Atlantic Packet Satellite Experiment, and the design of a local-access network to support speech and data experiments in the future wideband test-bed network.

Our LSI channel vocoder has been redesigned to use off-the-shelf microprocessor components instead of the original TMS9940 microcomputers, which have not been released by Texas Instruments. The new design features a novel implementation of the Gold-Rabiner pitch detector, based on logarithmic waveform features. It also exhibits a better form factor and lower power dissipation than the original. Work on the spectral envelope estimator vocoder has been focused on improving the synthesizer structure, both for reasons of computational complexity and improved voice quality. Five pitch-detection algorithms were informally tested under a variety of environmental situations, with the result that the Gold-Rabiner method performs best in the absence of acoustic noise or distortion, and the harmonic pitch detector seems to offer the best potential for use with telephone speech. An LSI chirp-Z transform device has been interfaced to an LDSP and is now ready to support experiments aimed at establishing the utility of CCD-based FFTs for speech processing applications. A stream capability has become available in the Atlantic Packet Satellite Experiment, and conferencing experiments

using the streams are proceeding in both the SATNET and internet environments. Performance does not yet meet expectations due to problems in host computer and SATNET software. A design has been formulated for the local voice access network, and a general voice terminal structure has been configured for use in that access area and in other packet speech network environments. Our initial pilot access net implementation will use an ETHERNET-like contention strategy for sharing a single cable among the various voice terminals and the speech concentrator.

INFORMATION PROCESSING TECHNIQUES PROGRAM

PACKET SPEECH SYSTEMS TECHNOLOGY

I. CCD/BELGARD VOCODER

A. Introduction and Summary

During the first quarter of FY 79, it had become increasingly apparent that the Texas Instruments TMS9940 microcomputer, a key element in the vocoder design, had encountered developmental difficulties, and questions arose as to the timely availability of the device. Consequently, it seemed advisable to re-assess the possibility of finding a viable alternative. As it happened, the TMS9940 was originally selected because it incorporated several unique and highly desirable features in its design. These included a true single-chip architecture, ample onboard program and data memory, a relatively high level of performance, a powerful instruction set (including an explicit multiply), and a 16-bit precision format. The latter three were considered particularly critical to the pitch extraction process. At the time the design was frozen, there were no other processor choices.

However, some timely conversations with representatives of Marconi Space and Defense Systems, the designers of a novel all-digital version of the Belgard, suggested that a system based on 8-bit microprocessor technology was indeed possible and capable of quite satisfactory performance.

Given these events as impetus, a feasibility study was undertaken directed at developing a second vocoder design based upon currently available, off-the-shelf, 8-bit microprocessor technology. The result, described in the following sections, has proven to be superior in several respects to the TMS9940-based version. An engineering prototype unit has been designed, fabricated, debugged, and subjected to preliminary testing. Not only has performance commensurate with the original design been demonstrated, but considerably better form factor and dissipation figures have been achieved.

B. Architecture

The new architecture (Fig. 1) retains all of the essential operational features of the original while at the same time substantially reducing hardware complexity. It is a full-duplex unit featuring four choices of transmission rate. Essential subsystems include a compact analog signal conditioner and a pair of minimum-configuration Intel 8085A-2 microprocessor complexes for pitch extraction and controller functions.

The analog subsystem (Fig. 2) features several improvements over the original including:

(1) Substitution of a 5th-order elliptic presampling filter for a 7th-order design.

(2) Reduction of the pitch detector data acquisition circuitry by a factor of 2.

(3) Elimination of explicit analog voicing decision hardware.

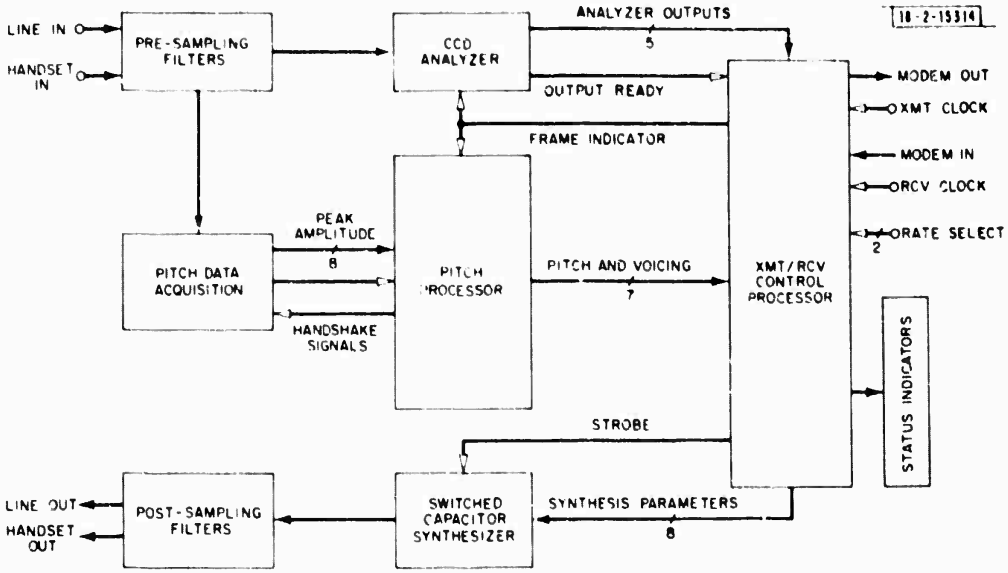


Fig. 1. NMOS vocoder architecture.

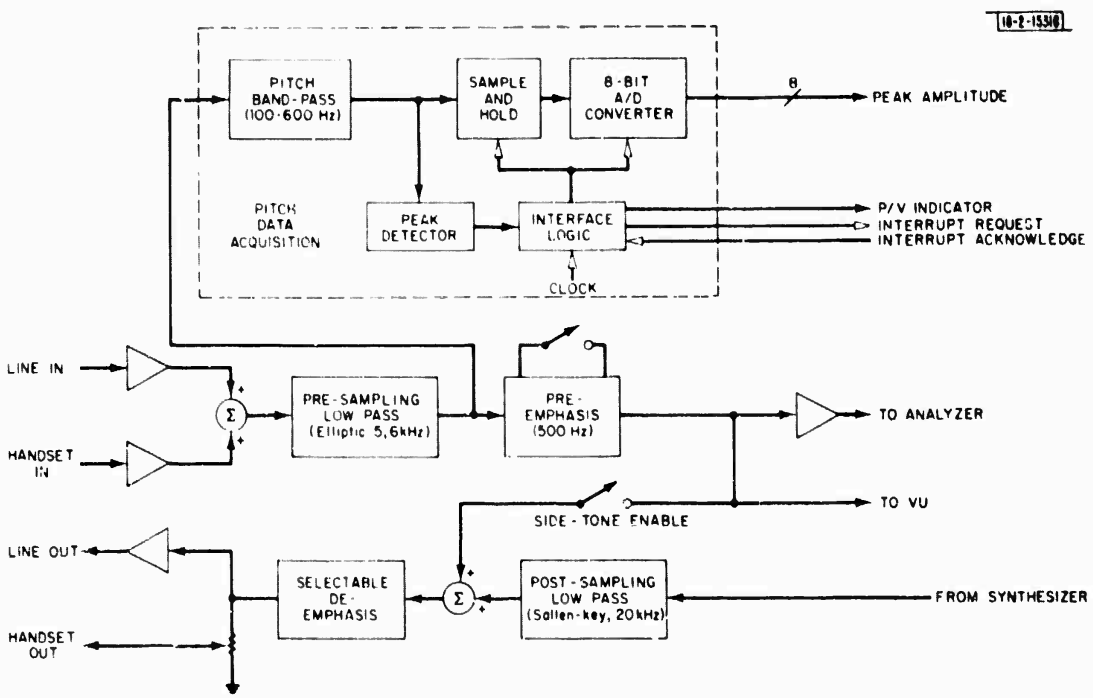


Fig. 2. Analog subsystem.

(4) Substitution of a single 8-bit A/D converter in the pitch detector for a pair of 12-bit units.

(5) Simplification of the pitch low-pass filtering by replacing the two Bessel structures with a single 3rd-order Butterworth design.

These changes, which are largely related to an alteration in pitch detection strategy, contribute significantly to an overall package count reduction of nearly a factor of 2 over the original design yet still require only strictly off-the-shelf parts.

In the following sections, the pitch detector and controller subsystems are described in some detail. It should be noted that although the new design requires only two processor complexes as opposed to the five of the original, there is no net real estate advantage to be gained from this quarter given the multiple-chip nature of the 8085 microprocessor family.

C. Pitch Extractor

The pitch extraction algorithm is a direct variant of the classic Gold method[1] as modified for real-time implementation.[2] The basic structure is shown in Fig. 3. This is a time-domain approach which seeks to obtain an indication of waveform periodicity by measuring the elapsed

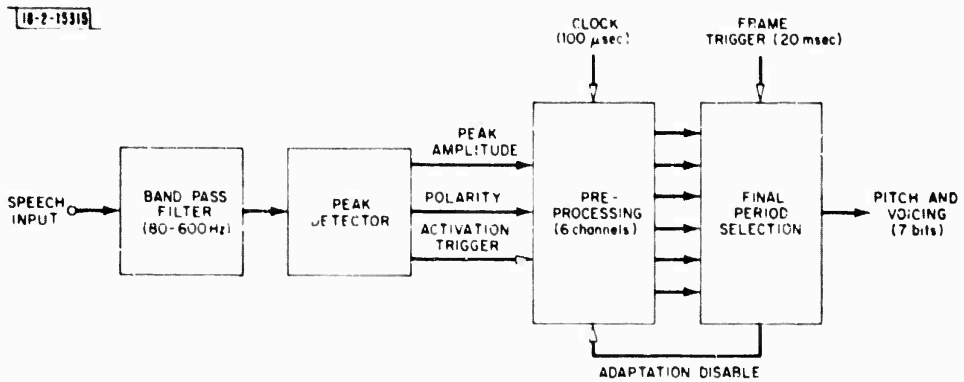


Fig. 3. Gold pitch detector.

time (in 100-μsec increments) between suitably gated points of inflection. Six-fold parallelism is exploited in a clever way to enhance the overall robustness of the periodicity measure. In addition to a period estimate, a confidence factor is produced as a by-product serving as a major basis for the voiced/unvoiced decision. Voicing and period determinations are typically required at 10- to 20-msec intervals.

Though the Gold technique is an established performer in relatively benign environments, it has not enjoyed widespread acceptance in the narrowband speech processing community at large. Aside from issues of robustness, this is largely due to its relatively high computational complexity, and a very waveform sensitive behavior which makes computation time estimates rather difficult. Various attempts to simplify the basic technique have been put forth by Gold, Rabiner, Bially, and others.[3] The pitch detector chosen for the original vocoder design was in fact a composite of these ideas as has been previously reported.[4]

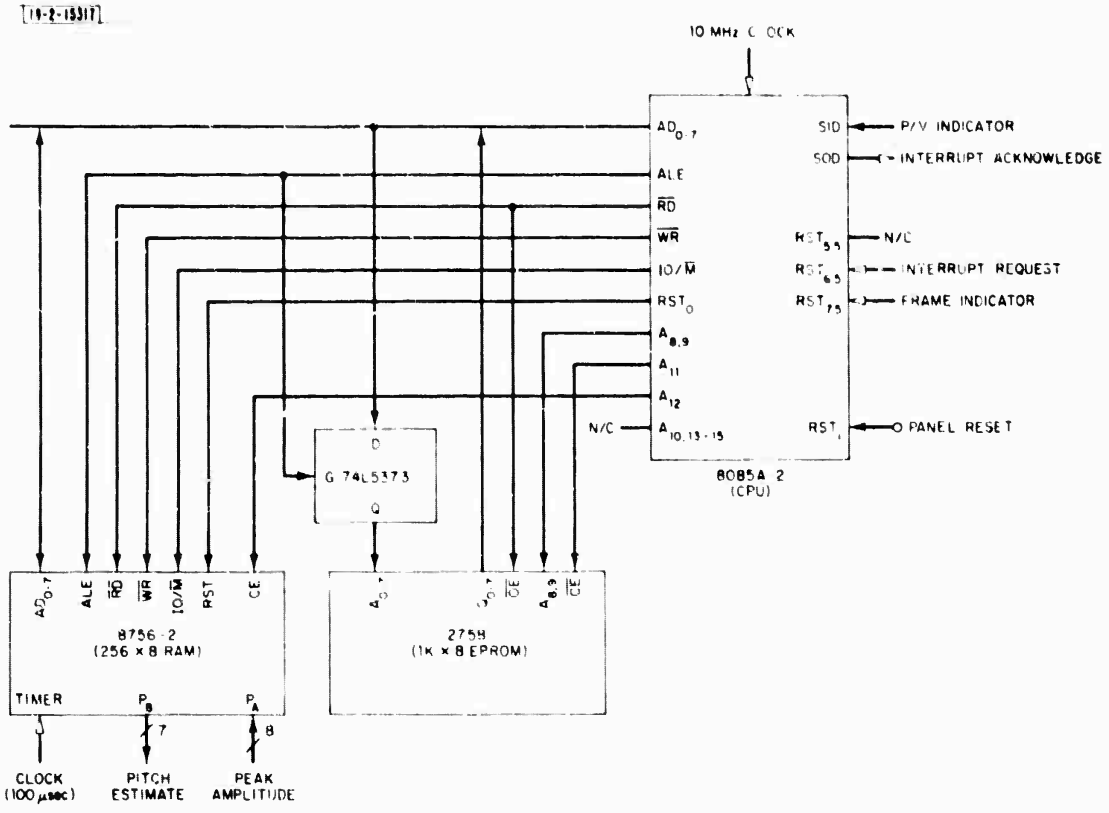


Fig. 4. Pitch processor.

However, recently a particularly ingenious implementation has been suggested[5,6] which not only drastically reduces overall computational complexity, but is compatible with an 8-bit precision format. Essential elements of the approach are:

(1) The use of logarithmic arithmetic to compute the threshold tests in the six peak-processing units. This technique eliminates explicit multiplications and deals with the exponential threshold in a natural way.

(2) Using a single adaptation parameter (P_{AV} in the literature) to control all six peak processors. This parameter is updated once per 20-msec frame based on the final pitch and voicing decision. In the classic approach, there are six separate controls updated each time a peak is accepted in a given peak processor.

(3) Implementation of an entirely new algorithm for final period selection and voicing. Rather than the complex coincidence checking philosophy of the classic method, a histogram strategy is used. A 64-place histogram is compiled from the standard 36 period measurements subsequent to recoding in a 6-bit approximate log format. After suitable smoothing of the histogram, the bin corresponding to the histogram peak is considered the winning candidate, and the peak amplitude is used as an indicator of the confidence of the estimate. Comparison of the peak with an empirically determined threshold yields the voicing decision.

Rigorous real-time simulation of this modified method has revealed, surprisingly enough, little or no performance degradation relative to either the classic technique or the two-channel design originally planned for implementation. Furthermore, the hardware required to realize this critical subsystem reduces to a small number of analog components (. . .) and a 10-MHz minimum configuration (3-chip) 8085A-2 microcomputer including 1K × 8 EPROM and 256 × 8 RAM memory complement (Fig. 4). In actuality, only 646 bytes of ROM and 104 bytes of RAM are required. Measured computation times are on the order of 15 msec per frame (worst case) which is equivalent to 75 percent of allotted real time.

D. Control Functions

The 8085A-2 microcomputer family has sufficient processing power, memory, and I/O capacity to allow integration of all command/control and interface functions into a single conceptual unit. Transmitter and receiver programs are arranged to coexist in a real-time-compatible manner which relies heavily upon the convenient interrupt structure of the microprocessor. The primary duties of the control processor are:

(1) Transmission rate determination and attendant initialization of all parameter encoding/decoding logic. This function is performed on power-up or in response to a system reset.

(2) Determination of frame boundaries, serial-link interfacing, and bit-stream synchronization.

(3) Encoding/decoding and formatting of data.

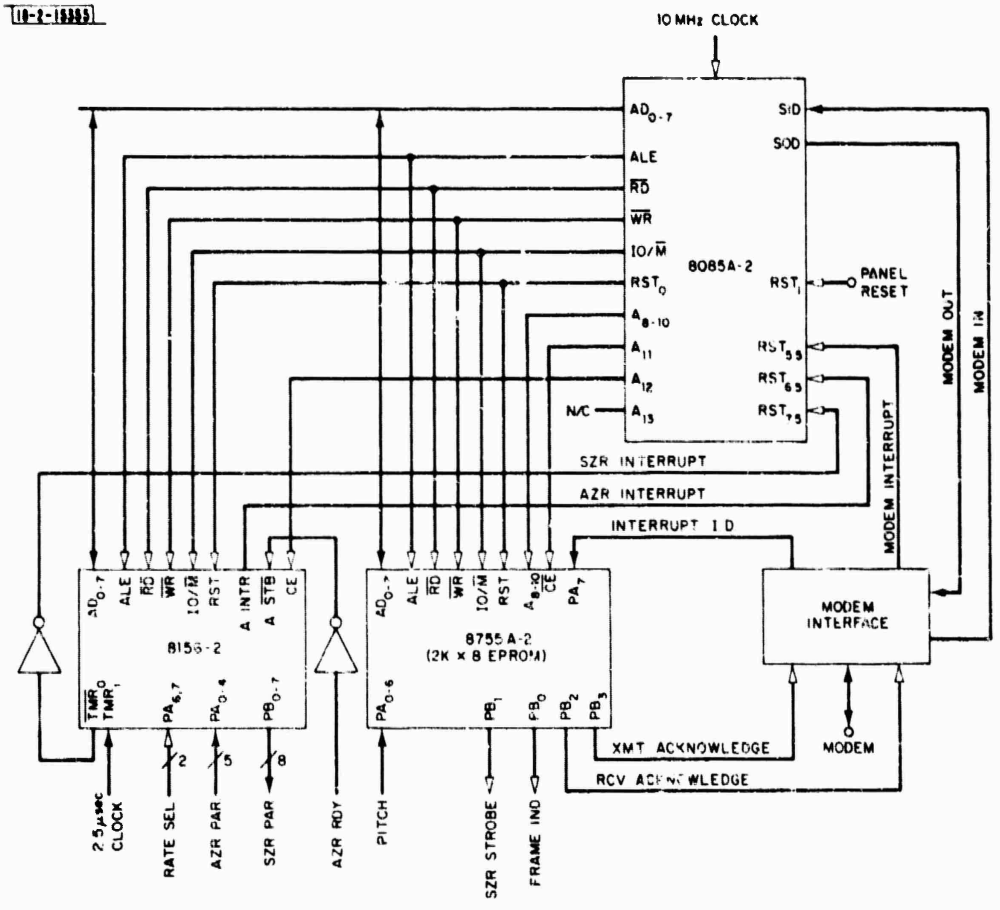


Fig. 5. Control processor.

(4) Interfacing with the custom analysis and synthesis devices.

(5) Command, control, and interface with the pitch extraction subsystem.

The architecture of the control processor is shown in Fig. 5. Serial-link interfacing is handled through special CPU ports and is interrupt driven. Communications with the analyzer and synthesizer are also conducted on an interrupt basis. The internal timer feature of the 8156-2 RAM/IO unit is used to assure a regular data transfer rate (1 msec/output) to the synthesizer. The 8156 and 8755 memory/IO devices have sufficient port capacity to accommodate the necessary pitch computer, analyzer, and synthesizer parallel interfaces. All told, the controller microcode requires 1174 bytes of EPROM, 170 bytes of RAM, and 14 msec per frame of run time at the 4800-bps (worst case) rate.

E. Design Status

The completed engineering prototype unit is comprised of 30 ICs and 15 discrete component carriers mounted on a 6-in. × 6-in. wire-wrap board (Fig. 6). It weighs 6.25 lb, occupies 0.12 ft^3, and dissipates 5.3 W. A test scenario has been configured using the LDSP facility to

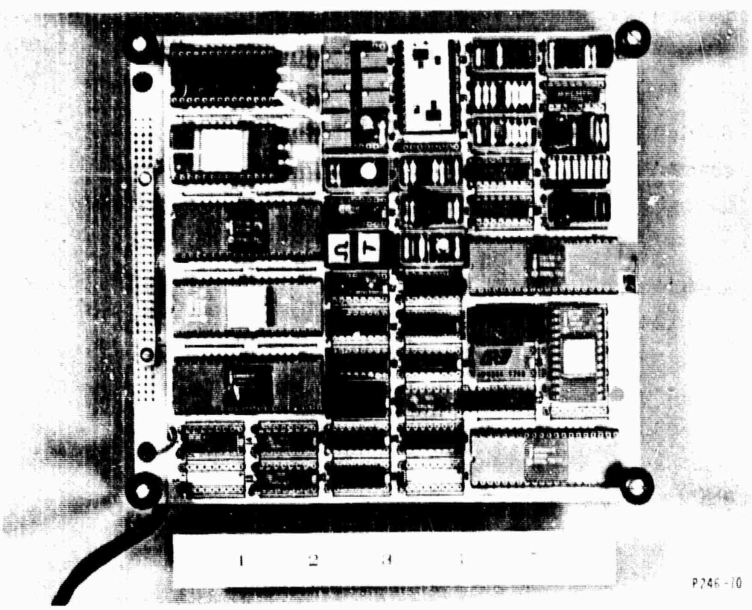


Fig. 6. Vocoder board.

perform real-time emulation of the analysis and synthesis functions. In this way, it was possible to establish the correctness of all control protocols as well as pitch extractor performance without requiring finished versions of the analyzer and synthesizer chips.

A second unit is nearing completion which will permit genuine, two-way communication tests to proceed as soon as finalized versions of the custom devices become available.

F. Remaining Work

It remains to perform detailed performance evaluation of the vocoder devices upon their arrival. It is tempting to also consider several relatively small system design refinements which may prove to be desirable in the future. Some possibilities are:

(1) Incorporating newly available switched-capacitor recursive filters in the analog presampling circuitry. Some power and parts count savings are possible here.

(2) Incorporation of new digitally programmable attenuators in the analog input path. This could serve as a microprocessor-controlled AGC increasing overall system dynamic range.

(3) Incorporation of TMS9940 microprocessors when (and if) they become available. Three units would be necessary, representing a savings of two large ICs over the present version.

(4) Revision of the design to provide suitable terminal interface data such as frame boundary and silence indicators.

(5) Incorporation of an improved 1200-bps coding algorithm. With some careful planning, it is very likely that the McLarnon method[7] can be made compatible with remaining RAM constraints in the controller. The current technique is known to be considerably inferior.

(6) Revision of the microcode to provide compatibility with the Belgard transmission and synchronization formats. This would allow a CCD/Belgard to talk to a Marconi Growler, at least at the 2400-bps rate, and might be a valuable demonstration vehicle.

(7) Development of a modified analyzer chip for robust pitch processing. Recent studies have shown that a filter bank-modulator-summation cascade is a useful structure for adaptively preconditioning speech corrupted by an acoustic noise background. This signal-to-noise ratio enhancement process impacts most dramatically on the pitch extractor subsystem which requires only the lowest 600 Hz of bandwidth for the Gold algorithm. Augmentation of the four or five lower order analyzer channels with a modulation-summation capability could provide a mechanism for implementing an optimally adaptive pitch low pass filter in addition to the usual analysis function. The chip design ramifications of such a structure need further study.

II. VOCODER STUDIES

A. Speaker-Adaptive Vocoder

The DRT intelligibility scores have been received from Dynastat. The scores are as follows [percent (std. dev.)]:

System / Environment	Gold-Rabiner Pitch	ML Pitch	ML Pitch and Noise Suppression
Clear	86.2 (0.68)	85.2 (0.74)	84.7 (0.66)
ABCP (dynamic mike)	70.6 (0.85)	76.8 (1.80)	78.3 (0.77)
ABCP (confidencer mike)	73.4 (1.28)	75.8 (1.67)	76.3 (1.14)

These scores suggest several observations. In the clear, the systems are essentially of equivalent intelligibility. In the airborne command post (ABCP) noise environments, the maximum likelihood (ML) pitch systems are clearly superior. The noise canceling (confidencer) microphone, which aids the Gold-Rabiner (GR) pitch system actually degrades the ML pitch systems. Finally, the noise suppression produces only insignificant intelligibility improvements.

The final observation appears surprising in the light of the large apparent improvement in the signal-to-noise ratio. It suggests that the human ear and brain can do as well or better than this algorithm in separating the speech from the noise. The noise-suppressed output is, however, far more pleasant to listen to than is the non-noise-suppressed output. Thus, the primary advantage of the noise suppression in this vocoder structure may be listener fatigue reduction.

An error in the previous system has been located and removed. The compensation for the analyzer gain had errors up to about 6 dB. The synthesizer impulse response interpolation has also been removed from unvoiced frames to allow better reproduction of stops.

We have spent a portion of this reporting period in attempting to refine the synthesizer portion of the spectral envelope vocoder. This work was motivated by the computational complexity of the minimum-phase convolutional structure, and by the opinion that part of the characteristic buzziness of homomorphic vocoders derives from the synthesis strategy. A property of the envelope estimation function is that it is discontinuous in its first derivative by virtue of the straight-line segments that are used for interpolation. A third-order spline, which is continuous up to and including the second derivative, has been tested as an alternative interpolation function in the estimator algorithm. (For comparison, the linear interpolation used in the current version may be viewed as a first-order spline.) This yielded only slightly clearer speech reproduction. It is computationally far more complex than the linear interpolator currently used and probably not worth the additional complexity.

The channel synthesizer, while an interesting research tool into the fundamental properties of channel synthesizers, proved to yield speech quality inferior to that of the direct-convolution synthesizer. Another synthesizer structure of interest is the LPC synthesizer. We are speculating that the spectrum envelope estimator can be used as an analyzer structure for generating LPC parameters. If coding were performed on the resultant LPC parameters instead of on the spectral envelope function directly, one could obtain a system that uses a conventional LPC synthesizer and is thus compatible with existing LPC devices. The analyzer should exhibit potentially better performance than an LPC analyzer by virtue of the strengths of the spectral envelope estimation process. We are currently pursuing this avenue of research.

B. Pitch Detectors

If we define a vocoder to be a device that exploits the convolutional model of speech, then it follows that implementation of such a device can be gracefully partitioned into two components. One is the spectral estimator and associated synthesizer; at present, three configurations of this component have received widespread attention: the channel vocoder, LPC, and spectral estimation and convolution based on the high-resolution spectrum (e.g., homomorphic). Not only have many variations of these algorithms been studied but, in addition, development of appropriate LSI to realize these components is being actively pursued.

The other component of a vocoder is the pitch extractor and associated voicing detector and excitation generator. LSI implementation of this component has not yet been as heavily attacked, to a great extent because there is less agreement on the relative merits of the various available pitch algorithms. Also, the pitch algorithm appears to be quite vulnerable to environmental degradation. Recently, five pitch algorithms developed at the Lincoln Laboratory were informally tested under a variety of environmental situations, listed below:

(1) Telephone speech.

(2) Paragraphs (15 to 30 sec long) read from a book or a newspaper (male and female speakers) using a close-talking dynamic microphone in a handset.

(3) Speech recorded from a pilot in an operating F-16 aircraft; the pilot wore a helmet and the microphone was inserted within an oxygen mask.

(4) Good-quality microphone recording with ABCP (airborne command post noise background) (S/N approximately 10 dB).

(5) Good-quality microphone recording with helicopter noise background.

(6) Same as (4) but with approximately 6 dB S/N.

(7) Same as (6) but with confidencer microphone.

(8) Confidencer microphone speech with noise removed.

(9) Same as (8) with good microphone.

Loosely speaking, our speech inputs can be classified as: (a) telephone speech, (b) speech of reasonably good quality, and (c) speech with a noisy environment.

Any quantitative comparison was considered impractical, since the five pitch algorithms were attached to four different spectral estimation algorithms. It was therefore agreed at the outset to listen informally and try to evaluate only the pitch performance. Since spectral degradation was not an issue, all systems were run in the uncoded mode. The nine different speech samples enumerated above were passed through the following five systems:

(1) The Gold-Rabiner pitch with a spectrally flattened channel vocoder.

(2) The harmonic pitch with the Belgard spectrum.

(3) The homomorphic pitch with LPC spectrum.

(4) The McAulay pitch detector (based on the principle of maximum likelihood) with LPC spectrum.

(5) The Paul pitch detector (based on spectral flattening of the speech prior to pitch estimation) with the spectral envelope estimator.

Since the master input tape contained about 20 min. of speech, the 5 systems corresponded to 100 min. of listening. Under these circumstances, judgments are necessarily loose and extreme caution is needed to interpret what was heard. Given these constraints, we now discuss the results.

For the undegraded speech (item 2), all pitch detectors performed satisfactorily. If a choice had to be made, the Gold-Rabiner detector would be chosen, perhaps because it is a time-waveform detector and can follow rapid variations. Similarly, the McAulay detector seemed to be of comparable crispness to Gold-Rabiner; its time constant was also short.

For telephone speech, the harmonic and homomorphic appeared to be the best. Since telephone speech often contains much phase distortion and tends to make the waveform less "peaky," Gold-Rabiner would be expected to suffer, and it did, often getting confused during voicing and calling it hiss.

For the noisy inputs, the two versions of the maximum-likelihood detectors proved, on the average, superior. Both these detectors can be said to have ability to track during noise, whereas the other detectors had no such inherent capability. Nevertheless, in many cases, we felt that the harmonic detector outshone Gold-Rabiner in noise.

It should also be emphasized that even in these difficult situations all the systems tested would probably permit sufficiently intelligible speech communication. Therefore, grounds exist for the LSI implementation of any of these detectors. However, bringing considerations other than the above informal listening into the picture, our best recommendation would be to pursue LSI activity on behalf of Gold-Rabiner and harmonic. The former has proved to be very reliable for the great majority of situations one would encounter, easy to implement, and yields "crisp" pitch, which we judge to be slightly more satisfactory (given that there are no errors) than detectors that utilize a greater degree of smoothing. The harmonic detector is also easy to implement and has proved to be quite robust. If the vocoder designer had the option of providing either a "harmonic chip" or a "Gold-Rabiner chip," or (better still) of providing both with a switch, a vast majority of users would have little cause to complain about the "pitch problem."

III. CCD CHIRP-Z TRANSFORM HARDWARE

The CCD Chirp-Z Transform peripheral has been expanded to provide either single-channel magnitude output or dual-channel I and Q vector outputs to the LDSP for off-line transform processing. The basic parameters of this peripheral are:

512-point transform

Sampling rate – 12.5 kHz

Computation rate – 4 msec

Analysis band – 0 → 62.5 kHz

Resolution BW – 244 Hz

Input options –

- Rectangular window (5601-1)
- Hanning window (5601-2)

Weighting coefficient accuracy – 8 bits + sign

Dynamic range – 60 dB

Output options –

Magnitude ($\sqrt{|I|^2 + |Q|^2}$)

Vector (I, Q)

Figure 7 shows a block diagram of the CZT hardware attached to the LDSP. Raw speech samples are collected in the LDSP using the signal conditioner. The samples are preprocessed by the LDSP as desired and passed to the CZT via a 12-bit D/A. A test option for direct analog inputs allows a CZT check independent of data acquisition hardware.

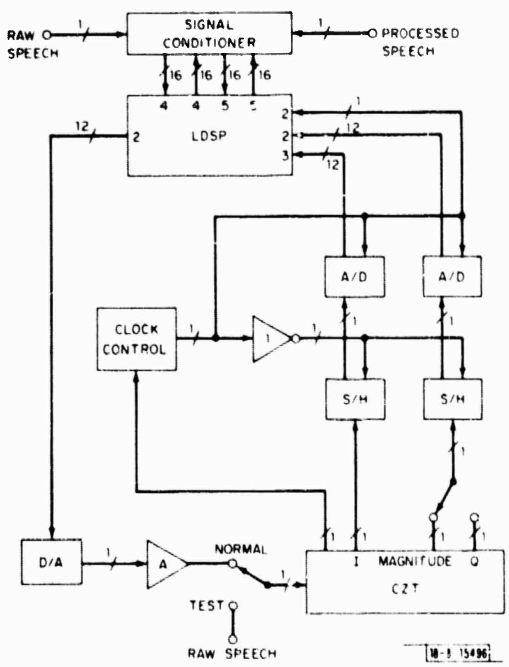


Fig. 7. CZT hardware.

Software has been written to perform block and sliding transforms in a test scenario. Candidate vocoder algorithms for system tests of the CZT hardware include the homomorphic vocoder and the spectral envelope estimation vocoder. Although present software was written for 256-point forward and reverse transforms, it can be modified to accommodate a 512-point transform. Initial experiments will involve the hardware magnitude option. The vector option will allow LDSP computation of more accurate magnitude and phase.

IV. SATELLITE AND INTERNETTED CONFERENCING

Linco'n Laboratory has provided hardware and software to support voice conferencing experiments as part of the ARPA Atlantic Packet Satellite Experiment. Four sets of speech hardware made up of a linear predictive (LPC) vocoder and interface equipment to connect to PDP-11 computers are in place and operational. Software to allow voice conferencing has been demonstrated using broadcast packet communication in an early version of the packet network (SATNET). Current work has been concerned with the modification of that software to use a newer version of the

SATNET which provides a broadcast stream capability which should be ideally suited to support voice conferencing. The stream represents a reservation of satellite channel capacity which can be shared among the sites participating in a conference. The stream is intended to provide packet communication with less average delay as well as less variance in delay than would be expected with other demand-access methods being considered for packet satellite use. The current status of the stream capability in SATNET is that it has just become available for use. Our conferencing program is operating using the stream capability in a mode in which speech is echoed from the satellite. Performance does not yet meet our expectations. Problems in the interfacing software in the PDP-11 host computer and/or in the SATNET are being investigated by Bolt Beranek and Newman, Inc., which has responsibility for those parts of the overall system.

Once SATNET conferencing is operating satisfactorily, we expect to proceed with debugging the software to support internetted conferencing between ARPANET and SATNET. All the necessary programs have been written and are awaiting debugging. Lincoln Laboratory and the Information Sciences Institute in Marina Del Rey, California, have implemented software versions of the LPC vocoder algorithms embodied in special-purpose hardware in SATNET. These two sites will be ARPANET participants in internet conference experiments. Special code has been prepared to run in the gateway machine between ARPANET and SATNET which will serve as a central control program (CHAIRMAN) in the ARPANET part of the conference and as one of the participants in the distributed-control SATNET part.

The approach being used in the ARPANET/SATNET experiment to realize internetted conferencing by interconnecting two rather different conferencing techniques is not particularly attractive as a general solution to internetted conferencing since it requires specialized actions on the part of the gateway between the nets. In the general case, gateways between each pair of networks would require software specialized to that particular pair of nets. A more general solution might be possible if something more than the presently defined datagram-based internet environment were provided by the basic gateways. We are working toward this more general solution by participating in the development of protocols appropriate to the handling of speech data and conferencing. The expected outcome will be a new Network Voice Protocol suited to the needs of the ARPA Internet Project as well as the wideband satellite experiments.

V. ACCESS-AREA DESIGN

The primary purpose of the local access area is to collect voice traffic from the individual voice terminals for transmission over the wideband satellite or terrestrial links and distribute incoming traffic back to these terminals. In addition, it would be desirable if the local access area could support other functions without interfering with the primary function. These additional functions include the collection of data traffic for transmission on the wideband network and the transmission of voice and data traffic within the local access area.

A number of possible implementations have been suggested. In this section, we summarize the major issues and trade-offs involved in selecting a particular implementation. A tentative set of requirements for the access area is given along with a list of the criteria which are important in evaluating a system. The various candidate systems are then briefly described, and the trade-offs in the various systems are listed. Finally, some simulation results are described and discussed.

A. Requirements

The local access area provides a means of connecting the voice terminals to each other and to a wideband transmission network as shown in Fig. 8. The individual terminals are connected to a common transmission medium which is time shared according to some control algorithm. A concentrator, also connected to the access area, reformats data for transmission on the networks.

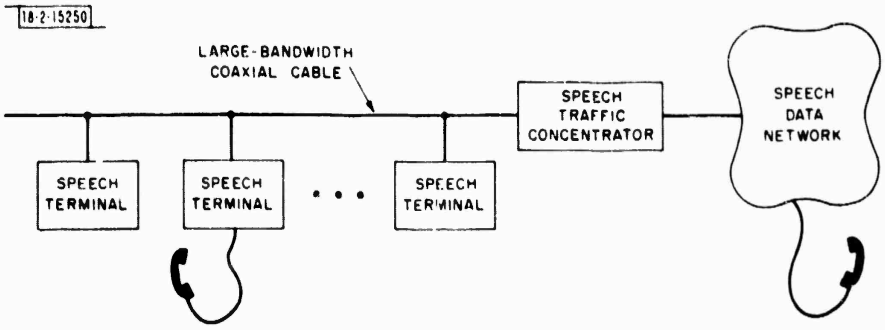


Fig. 8. Access-area geometry.

The local access network should provide reliable, low-cost, flexible digital communication among the different terminals. The network should be designed to meet the following requirements:

(1) The network must support voice terminals with different data rates, packet sizes, and packet rates corresponding to different vocoder implementations. The clocks of the different terminals are assumed to be unsynchronized with each other or with the network.

(2) The access area must support the signaling protocols required for call initiation, dialing, termination, and other control.

(3) The local access area must be able to support conferencing and broadcast message traffic.

(4) The access area must be able to support the control information necessary to realize future flow control algorithms.

(5) The system should support direct communication between terminals within the local area without intervention of the concentrator. This will be essential for system development when the system will have to operate without a concentrator and would be desirable in the final system.

(6) The resources of the channel should be allocated dynamically to take advantage of speaker activity.

(7) The network should be compatible with the protocols required for end-to-end encryption.

(8) The local access network should be able to handle data as well as voice. The access protocol should be able to assign different priorities to voice traffic, interactive data traffic, and file transfers.

Our present goal is to develop and construct a small-scale pilot system that provides access to a local host (mini-concentrator) for several voice terminals. The pilot system will be used to develop and demonstrate the techniques that will be required in an eventual full-scale implementation, in addition to serving as a vehicle for voice terminal and speech concentrator experiments.

The full-scale local access area is assumed to have the following characteristics:

(1) The cable data rate is assumed to be 1 Mbps. This is large enough to support a meaningful number of users and provide a significant load to the wideband network, but not so large that the cable transmission represents a significant technical development. Integrated circuits are available which support cable interface protocols at these rates.

(2) The network will be designed so that extension to 1000 terminals physically connected is possible. This means that the transmission protocols must be capable of supporting this number and that the hardware implications of such an extension must be thoroughly examined.

(3) Terminals will be located within 1 km of the concentrator. The implications of going further should be examined.

(4) The network should support 50 to 100 off-hook voice terminals. The table below shows the maximum number of voice terminals which could be supported on a 1-Mbit cable if there were no packet overhead or cable access overhead.

Vocoder Rate (bits/sec)	Maximum Users
2,400	416
4,800	208
9,600	104
16,000	62
32,000	31

Besides these required features, there are several others which would be desirable in an operational system but are not necessary in an experimental system.

(1) Compatibility with other uses. (Certain configurations of the local access net could operate using the hardware of a CATV system. This would allow the transmission medium to be shared with other uses on a frequency-multiplexed basis.)

(2) Ability to connect or disconnect terminals without interrupting service to other users.

B. Evaluation Criteria

There are a number of trade-offs that can be made in a system like the one under consideration. An initial list of criteria is presented here roughly in order of decreasing importance.

(1) Capacity and delay – For a fixed cable bandwidth, there is a trade-off between traffic load and delay. The delay introduced in the local network should be small compared with the tolerable voice delay.

(2) Stability – The system should degrade smoothly as the traffic load increases.

(3) Number of terminals supported – The limits may derive from the physical connection or the protocols or both.

(4) Robustness – A fault at an individual terminal should not disrupt the network performance.

(5) Extendability – Some local net architectures lend themselves to expansion more easily than others.

(6) Cost and complexity.

There are new issues which arise when evaluating the application of existing distributed-data network architectures designed for data for possible application to the voice problem.

(1) Voice statistics – An off-hook terminal will generate packets regularly at the frame rate of the vocoder, at least during talkspurts.

(2) The trade-off between speed and accuracy is biased heavily in favor of speed in the voice network, while it favors accuracy in the data network.

(3) In local data networks the flow tends to be truly distributed, while in this voice network the concentrator tends to provide a focus for the traffic.

C. Candidate Systems

A number of candidate systems have been studied and evaluated. We have generally concluded that any of the architectures considered would meet the basic system requirements. Depending on particular system requirements, one or another system might be appropriate for a particular application.

For this reason, it was decided to try to make the terminal flexible so that it could be used with a number of different local-access architectures. For an initial demonstration, we will implement the single-cable baseband ETHERNET, but we will retain sufficient flexibility in the hardware to support future experiments with the others. Some of the more attractive options are discussed below.

1. ETHERNET

Two versions of the ETHERNET are currently being considered. The first is the basic baseband network described by Metcalf and Boggs.[8] In this configuration, all the terminals are connected to a common cable which is used for reception and transmission as shown in Fig. 9.

Transmission works in the following way. A terminal with a packet to send, first listens to determine if the cable is idle. It then decides whether to send in the next time slot according to some probabilistic adaptive algorithm. If a collision occurs, all terminals which are transmitting cease, and attempt retransmission in the next time slot, again with a probabilistic rule. The probability of transmission is adjusted based on the success or failure of the terminals'

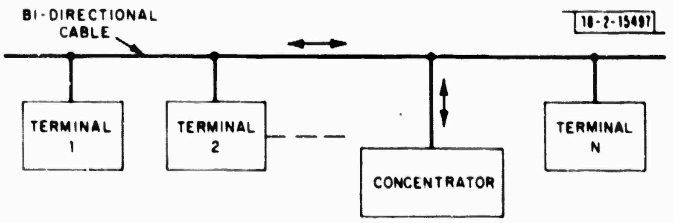


Fig. 9. Single-cable ETHERNET.

attempts. Each terminal adapts independently based on its view of the cable activity. The details of the algorithms and the results of our simulation studies are described in the appendix.

Collisions can be detected by monitoring the cable. The transmitter puts out a high level in either the first half or the second half of a bit slot depending on whether a zero or a one is being sent. By looking for activity in the cable during the other half of the bit time, the presence of other transmissions can be detected.

The second version of the ETHERNET is the one developed by MITRE in their MITRIX system.[9] In this system two cables are used, one for transmission, one for reception as shown in Fig. 10. The signal is transmitted on a carrier which makes it compatible with standard CATV hardware. The transmit procedure is analogous to the single-cable case. A terminal checks that the receive cable is idle before it begins to transmit. It detects a collision by listening for its transmission to be correctly received on the receive cable.

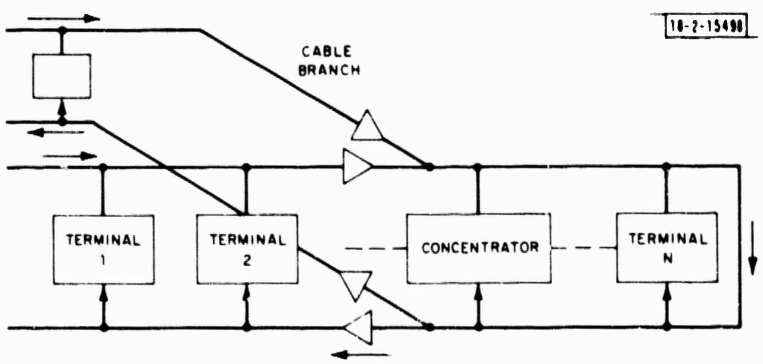


Fig. 10. Two-cable ETHERNET.

The system appears to have three major advantages. First, it can use all the transmission hardware designed for cable TV such as cable, amplifiers, cable taps, and the like. Second, since the transmission is unidirectional on each cable, it is easy to insert repeaters on long runs or to split signal onto two cables. It also allows the cable to be shared with other activities. Cable systems with several hundred megahertz of bandwidth are possible. A disadvantage is that the carrier modem is more complicated because of the high carrier frequency and the requirement for monitoring the receive data during transmission.

2. Ring Network

Another architecture which has been considered is the ring network. In this network, the terminals are connected as shown in Fig. 11. Transmission on the cable is unidirectional. Each

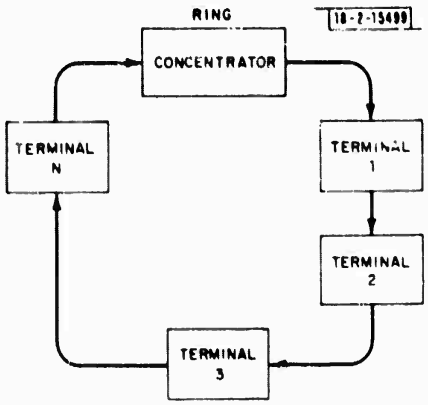


Fig. 11. Ring network.

terminal receives the data traffic and relays it to the next terminal. Access is controlled by passing a special control message or "token" from terminal to terminal around the ring. A terminal with a packet to send waits until it has received the control token. Instead of relaying the token to the next terminal, it transmits its packet and then follows it by the control token. The packet proceeds around the network, is read by each terminal including the intended receiver, and is finally removed when it gets back to the sender.

The advantage of the ring network is that the problems of packet collision are avoided by the control mechanism.

On the other hand, there are several disadvantages. Relaying the packet at each terminal introduces a small but cumulative delay as well as a reliability problem. Also, every terminal must remain active, at least for relaying messages, as long as the network is in use.

2. Centralized Control Algorithms

A centralized control architecture has been considered in which a central terminal would poll groups of terminals according to an efficient polling algorithm. Some simulations were tried, but no significant advantage in performance was found to offset the added complexity.

VI. VOICE-TERMINAL ARCHITECTURE

A voice terminal is that component of a speech network that comes into direct contact with the user. In the commercial telephone system, the terminal is a conventional telephone instrument, consisting of a pair of transducers (microphone, speaker) and dial up and ringing mechanisms.

For packet speech applications, a terminal should include speech digitization (vocoding) and packetization and depacketization functions at a minimum, in addition to the basic dialing, ring, and acoustic interfacing features. It also should include an appropriate interface (hardware and software) to whatever local network it is connected to. A more complex terminal may contain a complete set of network protocols that allow it to independently establish communication between itself and other terminals in remote access areas. Alternatively, concentrators in the local access nets can provide voice protocol services to groups of terminals of limited capability. In this section, we report on work directed at the design and implementation of a flexible voice terminal that can assume a variety of experimental configurations. Our objectives is to create a design that can interface to our pilot local access network or to a packet radio, while accommodating

any of a broad class of speech digitizers. Voice protocol functions will be programmable in the terminal software, allowing us to implement and experiment with various functional configurations.

Our design of a flexible voice terminal is based on several major considerations. First, we require a transmission protocol that is suitable for use with all candidate access area architectures. Second, the interface functions should be partitioned to separate those which are access area independent from those which are not. Such a breakdown is shown in Fig. 12. Finally, we would like to partition the terminal such that the same type of independence that is maintained with regard to different access areas is also maintained with respect to different vocoder types.

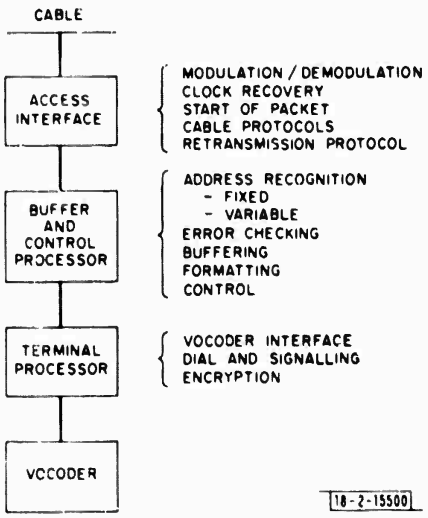


Fig. 12. Partitioning of terminal.

The Access Interface contains all the functions which are dependent on the specific architecture of the access area. It performs the modulation and demodulation of the signal and records bit time and start-of-packet information from the received signal. It also monitors the cable activity, decides when the cable is available, and when to start the transmission. In contention-based systems, it also determines that a collision has occurred and decides when to attempt a retransmission.

The buffer and control processor performs the architecture-independent transmission functions. For a transmit packet, the processor buffers the packet coming from the terminal processor; adds the starting synchronization frame, error check, and end-of-message frame; and transmits the packet to the access control unit when the cable becomes available. On reception, each packet is buffered, and the error check block is tested. The buffer and control processor checks the destination address of the received packet. If the packet is addressed to the terminal, then the packet is passed on to the terminal processor. Packets addressed to other terminals are discarded.

The terminal processor acts as the interface to the vocoder and also controls the dial and signaling protocols. The packet voice protocols are also formed at this point. The necessary computations for implementation of privacy algorithms for those users requiring secure communications, would also be carried out in the terminal processor portion of the voice terminal.

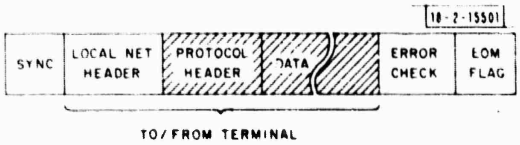


Fig. 13. Local-area voice packet format.

The format of the resulting packet is shown in Fig. 13. From the point of view of the access-area interface, the interface system delivers packets from one terminal to another. The protocol used within the packet is of no real concern to the access area except that the destination address may be at some specified position. Every received packet is buffered and its address is checked. Packets destined for other terminals are discarded. Note that if the terminal is configured as a stand-alone network host, the protocol header will probably contain complete voice and internet information. On the other hand, if the terminal is configured to act in conjunction with a concentrator that provides most of its protocol support, then the header will be much simpler. In either case, the access-area processor deals only with the local-net header and ignores the contents of the shaded portion of Fig. 13.

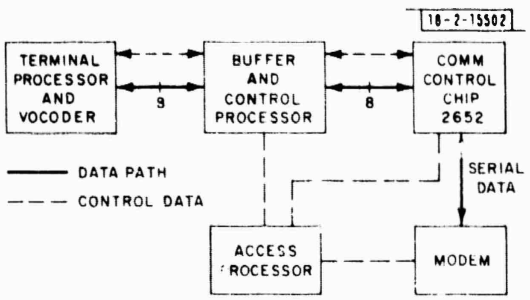


Fig. 14. Preliminary interface design block diagram.

A preliminary design of the access-area processor portion of a terminal for a single-cable ETHERNET system has been formulated. A block diagram of the system is shown in Fig. 14. The design takes advantage of the Signetics 2652 Communications Control Chip. This chip controls all the line protocols for a serial-data communications system at rates up to 2 Mbps. The modem transmits and receives a baseband signal over the cable. It is controlled by the Access Control Processor which is designed using a three-chip Intel 8085 microprocessor system. The buffer and control processor is also based on the Intel 8085, but includes DMA parts for fast program-independent block transfers of data to and from the Communications Control Chip and Terminal Processor.

APPENDIX
ACCESS-AREA SIMULATIONS

In a previous report,[10] we presented three possible adaptive-contention algorithms and simulation results for one of the algorithms. The simulation program has been extended to cover all three cases.

The optimum strategy, derived by Metcalf and Boggs,[8] calls for each terminal with a packet to transmit in the next time slot with a probability $P = 1/n$, where n is the number of terminals with packets to send. Unfortunately, each terminal does not know how many other terminals are trying to transmit, but can only estimate the number based on its observation of the channel activity. Call this estimate $\hat{n}$. Each terminal starts with an estimate $\hat{n} = 1$. In each time slot, a terminal with a packet listens to be sure the cable is idle and then tries to transmit with probability $P = 1/\hat{n}$.

Depending on the outcome of the attempt, each terminal will adjust its estimate.

Three different algorithms for estimating the traffic load were tested. The three are summarized in Table I. The terminals operate on a time slot equal to the time required to detect and respond to a collision on the cable. This time is roughly equal to twice the round-trip delay time on the medium. In each time slot when the cable is not busy, every terminal with a packet to send decides according to the probabilistic rule whether or not to transmit.

TABLE I
ADAPTIVE TRAFFIC ESTIMATION ALGORITHMS

	Algorithm A		Algorithm B		Algorithm C	
	Terminal		Terminal		Terminal	
Channel	Idle	Transmits	Idle	Transmits	Idle	Transmits
Idle	$\hat{n} - 1 \rightarrow \hat{n}$		$\hat{n} - 1 \rightarrow \hat{n}$		$\hat{n} - 1 \rightarrow \hat{n}$	
Packet sent successfully	No change	Packet sent No change	$\hat{n} + 1 \rightarrow \hat{n}$	Packet sent No change	No change	Packet sent No change
Collision	No change	$\hat{n} + 1 \rightarrow \hat{n}$	$\hat{n} + 1 \rightarrow \hat{n}$	$\hat{n} + 1 \rightarrow \hat{n}$	$\hat{n} + 1 \rightarrow \hat{n}$	$\hat{n} + 1 \rightarrow \hat{n}$

As seen from the point of view of a particular terminal on the access net, there are a total of five possible outcomes in each time slot. The terminal can either remain idle or transmit a packet. If the particular terminal under consideration remains idle, then the possible outcomes on the channel are:

(1) No other terminals transmit and the channel remains idle;

(2) Exactly one other terminal transmits, and a packet is sent successfully on the channel;

(3) More than one other terminal transmits, and a collision occurs on the channel.

If the terminal of concern transmits a packet, then the possible outcomes are:

(4) No other terminal transmits, and the packet is sent successfully;

(5) One or more additional terminals transmit, and a collision occurs on the channel.

Table I shows how the terminal would adapt its estimate $\hat{n}$ of channel activity for each of the five outcomes listed above, under the three algorithms considered. The algorithms are generally rather similar. For all three algorithms, the estimate is decremented by one when the channel is idle. If the terminal transmits a packet successfully, the estimate is left unchanged. If the terminal encounters a collision, the estimate is incremented by one. The differences between the three algorithms involve what happens in the cases (2) and (3), when the terminal remains idle and either a packet transmission or a collision occurs on the channel.

Algorithms A and B are both straightforward to implement since the idle terminal must only observe whether there is activity on the cable during the time slot. Algorithm C requires that an idle terminal distinguish between a successful packet transmission and a collision between two other terminals. This is more difficult than just detecting activity but it may be possible by measuring the duration of the activity.

A terminal with no packet to send does not adjust its estimate. When a terminal generates a new packet, it starts with an estimate equal to that with which it made its last successful transmission. It is in this way that the algorithm takes advantage of the periodic nature of the speech traffic.

TABLE II
SIMULATION PARAMETERS

	Case I	Case II	Case III
Vocoder Data Rate	2,400	2,400	16,000
Vocoder Frame Time (msec)	20	40	20
Data Bits/Packet	48	96	320
Total Packet Length	100	140	360
Cable Data Rate	5,000	3,500	16,000
Maximum Terminals in 1 Mbit (50% duty cycle, including expected contention overhead)	260	411	96

The three algorithms were each simulated for three different sets of vocoders with packet parameters shown in Table II. The first two cases are representative of a 2400-bps LPC vocoder; the third a 16-kbps CVSD vocoder. Each packet is augmented by a header of about 50 bits. Cases I and II can be thought of as the same vocoder but with two vocoder parcels combined into a single packet to increase the channel efficiency (but with increased packet delay). (There was some quantizing of the parameters due to the idiosyncracies of the simulation.)

The simulations for the most part showed very short average delays. However, one of the important characteristics of a system is the effect of delays much longer than the mean. In order to determine the effect of these infrequent, but large, delays, the simulation discarded packets which were not transmitted within one frame time. A count of discarded packets was kept. It is felt that for speech traffic a loss of 1/2 percent is generally acceptable, but the access area should not contribute to this loss. A conservative approach is therefore appropriate.

The results of the simulations are plotted in Figs. 15, 16, and 17 for the three cases. Each of the three algorithms was simulated at four different levels of activity. The number of vocoders was selected to give activity levels which were approximately 70, 80, 90, and 100 percent of the activity bound derived earlier.[10] The highest load is shown in Table II.

The plots show the transmission average delay for the packet transmitted vs the fractional channel utilization. The channel utilization represents the fraction of the time that information (either voice data or header) is being successfully transmitted on the channel. No information is being successfully transmitted when all terminals with packets are idle or when contentions are being resolved. Each point represents 60 sec of simulated activity or about 3000 vocoder parcels times for each speech terminal. For cases in which packets were discarded, the percentage lost is indicated for each point. Where no number is indicated, no packets were lost.

The results show that all three algorithms have substantially the same performance at low channel utilizations. At higher utilizations, algorithm A exhibits large delays. The delays start to grow and the fraction of packets lost becomes excessive. Algorithms B and C give essentially identical performance except for Case III in which B has a slight edge. It is not clear that the advantage is statistically significant. Previous analysis[10] indicated that C should have a slight advantage.

No stability problems were exhibited by any of the algorithms for the simulations run.

To compare these results with what might be obtained with a fixed TDMA system, the channel utilization should be discounted further by the percentage of overhead associated with the packet header. This is as high as 50 percent in Case I and as low as 11 percent for Case III. A TDMA system has a maximum efficiency of 50 percent because of its inability to take advantage of speaker silence intervals. This also must be discounted by some percentage to allow for control overhead and timing uncertainties.

The conclusion of the simulations is that the distributed access area is capable of supporting the traffic required in the access area. If we require that the local access network make a negligible contribution to the packet loss, say less than 0.05 percent, then the efficiency of the network is about 60 percent with the shorter packets and increases to better than 70 percent with the longer packets. The distributed access area is competitive with a fixed TDMA for a given cable bandwidth because it can take advantage of the speaker silence intervals. In addition, it gives added flexibility to accommodate different terminal types easily.

Of the possible contention algorithms, B exhibits the best performance. It is also simpler to implement than C because it does not require that a terminal distinguish between a packet transmission and a collision between two other terminals.

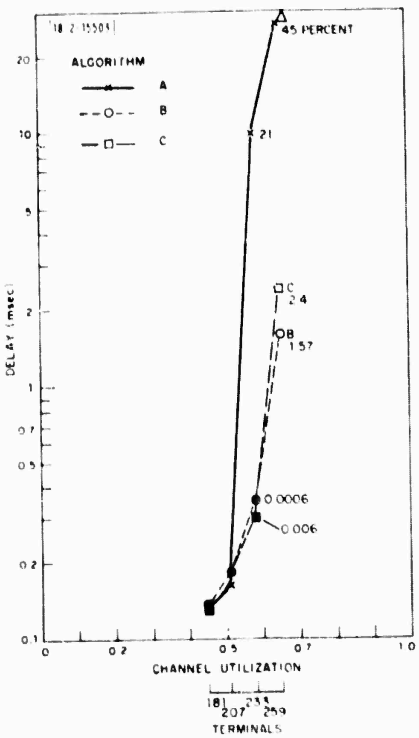


Fig. 15. Case I: 5,000 bps, 20-msec frame, 100 bits/packet.

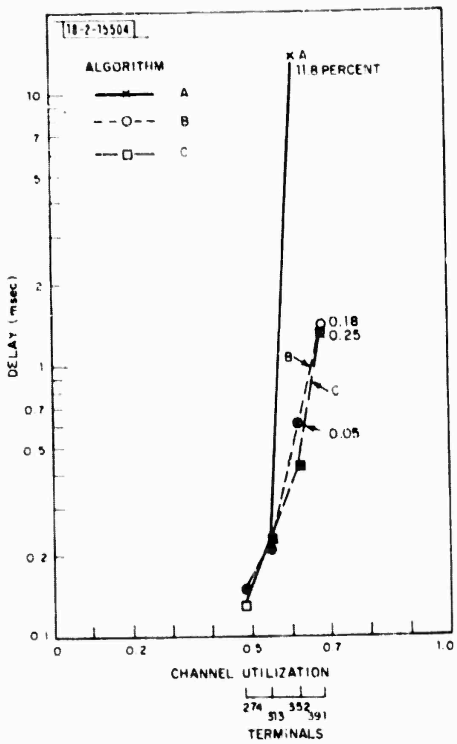


Fig. 16. Case II: 3,500 bps, 40-msec frame, 140 bits/packet.

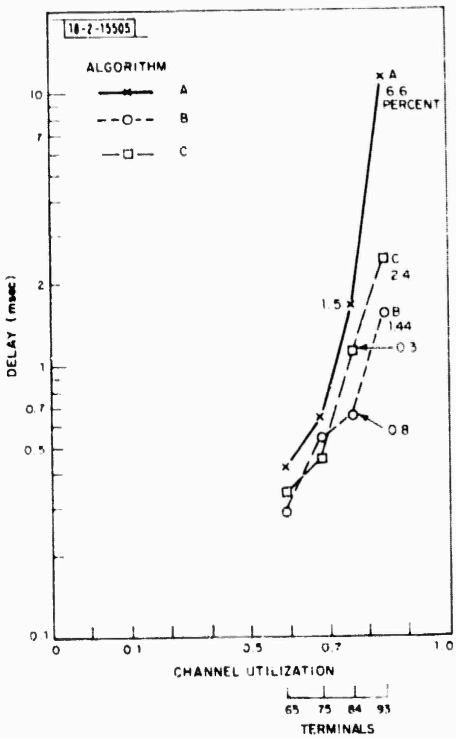


Fig. 17. Case III: 18,000 bps, 20-msec frame, 360 bits/packet.

REFERENCES

1. B. Gold, "A Computer Program for Pitch Extraction," J. Acoust. Soc. Am. 34, 916 (1962).
2. M. Malpass, "The Gold-Rabiner Pitch Detector In a Real-Time Environment," EASCON '75 Record (29 September 1975), pp. 31-A, G.
3. B. Gold, L. Rabiner, "Parallel Processing Techniques for Estimating Pitch Periods of Speech in the Time Domain," J. Acoust. Soc. Am. 46, 442 (1969).
4. Information Processing Techniques Program Semiannual Technical Summary, Volume I: Packet Speech/Acoustic Convolvers, Lincoln Laboratory, M.I.T. (31 March 1978), DDC AD-B028561-L.
5. W. Amos, N. Kingsbury, Marconi Space and Defense Systems, private communication (October 1978).
6. L. Kelly, N. Green, Communications Security Group, British Government, private communication (October 1978).
7. E. McLarnon, "A Method for Reducing the Transmission Rate of a Channel Vocoder by Using Frame Interpolation," ICASSP Record (April 1978), pp. 458-461.
8. R. M. Metcalf and D. R. Boggs, "ETHERNET: Distributed Packet Switching for Local Computer Networks," Commun. ACM 19, 395-404 (1976).
9. J. C. Naylor, "Data Bus Design Concepts, Issues and Prospects," 1978 EASCON Convention Record, pp. 34-39.
10. Information Processing Techniques Program Semiannual Technical Summary, Vol. II: Wideband Integrated Voice/Data Technology, Lincoln Laboratory, M.I.T. (30 September 1978), DDC AD-A067355.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ESD-TR-79-73
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)*: Information Processing Techniques Program Volume I: Packet Speech Systems Technology
5. TYPE OF REPORT & PERIOD COVERED: Semiannual Technical Summary, 1 October 1978 – 31 March 1979
6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Theodore Bially
8. CONTRACT OR GRANT NUMBER(s): F19628-78-C-0002
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Lincoln Laboratory, M.I.T., P.O. Box 73, Lexington, MA 02173
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: ARPA Order 2006, Program Element No. 62706E, Project No. 9P10
11. CONTROLLING OFFICE NAME AND ADDRESS: Defense Advanced Research Projects Agency, 1400 Wilson Boulevard, Arlington, VA 22209
12. REPORT DATE: 31 March 1979
13. NUMBER OF PAGES: 36
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*: Electronic Systems Division, Hanscom AFB, Bedford, MA 01731
15. SECURITY CLASS. *(of this report)*: Unclassified
15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT *(of this Report)*: Approved for public release; distribution unlimited.
17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*
18. SUPPLEMENTARY NOTES: Supplement to ESD-TR-79-81 (Vol. II)
19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*: packet speech; network speech; voice conferencing; ARPANET; SATNET; homomorphic vocoding
20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report describes work performed on the Packet Speech Systems Technology Program sponsored by the Information Processing Techniques Office of the Defense Advanced Research Projects Agency during the period 1 October 1978 through 31 March 1979.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

207654